FREE QA GUIDE

Your First 90 Days as a Software Tester

A Structured Approach to Early Career Success

By **Phillip Bailey** · 30-year QA veteran (startups to Fortune 500)

Table of Contents

1. Introduction: Setting Yourself Up for Success
2. Days 1-30: Foundation and Learning
3. Days 31-60: Building Competence and Confidence
4. Days 61-90: Establishing Your Reputation
5. The Top 5 Challenges You'll Face
6. Overcoming Common Obstacles
7. Building Relationships That Matter
8. Your 90-Day Success Checklist

Introduction: Setting Yourself Up for Success

Congratulations on landing your first QA role! The first 90 days in any new position are critical for establishing yourself as a valuable team member and setting the foundation for long-term success. This guide provides a structured approach to navigating your first three months as a software tester.

Why the First 90 Days Matter

First Impressions: Your colleagues and managers will form lasting opinions about your capabilities, reliability, and work ethic during this period.

Learning Curve: You may learn more new information in these 90 days than in any other period of your career. A new industry, new tools, new methods - ABSORB!

Relationship Building: The professional relationships you establish now can impact your success for years to come. Find your

Confidence Building: Successfully navigating these first months will give you the confidence to tackle bigger challenges. You got this - prove it to yourself.

Your Success Framework

This guide is organized around a proven framework that balances:

- **Learning and skill development**
- **Relationship building and collaboration**

- **Demonstrating value and reliability**
- **Overcoming common challenges**

Setting Realistic Expectations

Remember that becoming proficient in QA takes time. Your goal for the first 90 days isn't to become an expert - it's to establish yourself as someone who:

- Learns quickly and isn't afraid to ask questions or
- Takes initiative and follows through on commitments
- Collaborates effectively with team members AND can work without supervision
- Demonstrates attention to detail and quality focus

Days 1-30: Foundation and Learning

Your first month is all about absorbing information, understanding your environment, and establishing good habits.

Week 1: Orientation and Initial Setup

Day 1-2: Administrative Setup

- Complete all HR and administrative requirements - ensure you are set up with your new company

- Set up your workspace and development environment - This is where you make your money, so set yourself up for success
- Get access to all necessary tools and systems - What tools do you need? What do you need access to? Your team lead or colleague can help get you started
- Meet your immediate team members and key stakeholders - who will you be working with, report to, go to for help?

Day 3-5: Understanding Your Role

- Review your job description and expectations with your manager - easist to meet expectations if you understand them
- Understand your team's current projects and priorities - what's going on and how do you fit in?
- Learn about the products or systems you'll be testing - what tools do you need to learn first?
- Identify your initial training plan and resources - is it just "do what the tests say"? Or will there be formal training?

Key Actions for Week 1:

- ☐ Schedule one-on-one meetings with each team member if possible
- ☐ Create a learning journal to track new concepts and insights - review this regularly
- ☐ Set up your testing environment and verify access to all tools - ask for what you need and verify your tools are working
- ☐ Ask your manager about success metrics and expectations

Week 2: Learning the Basics

Focus Areas:

- **Company and Product Knowledge**: Understand what your company does and how your product fits into the market
- **Team Processes**: Learn how your team plans, executes, and reports on testing activities - and where you fit in
- **Tools and Systems**: Become familiar with bug tracking, test management, and communication tools
- **Documentation**: Review existing test cases, procedures, and quality standards as these will establish your first knowledge baseline

Daily Activities:

- Spend 2-3 hours reading documentation and exploring the product - this can be following established tests or whatever your lead directs you
- Shadow experienced team members during their testing activities whenever allowed
- Practice using the tools you'll need for daily work - the more proficient you are the more efficient and effective you will be
- Ask questions and take detailed notes - you will have fewer questions in the future if you do this

Key Actions for Week 2:

- ☐ Complete any formal training programs or courses
- ☐ Review at least 20 existing bug reports to understand the expected format and see variations

- ☐ Explore the product as an end user to understand its functionality and what it offers
- ☐ Create a glossary of terms and acronyms specific to your company/product - Note: All companies use their own terminology and all think that they are right

Week 3: Hands-On Practice

Focus Areas:

- **Guided Testing**: You should be getting experienced with following stories and test cases to execute testing
- **Tool Proficiency**: Using bug tracking and test management tools should become second nature
- **Communication**: Start participating in team meetings and discussions if you haven't already - ask questions, provide valuable feedback
- **Process Understanding**: Learn your team's specific workflows and procedures and ask why so that you understand their methodology

Daily Activities:

- Execute simple test cases with guidance from senior team members as needed
- Practice writing bug reports and get feedback on your work - don't just write the bugs, ensure they are reviewed until you feel confident
- Attend all team meetings and ask clarifying questions - you may not be the only one that needs clarification, but most people don't ask for it at work

- Begin building relationships with developers and other stakeholders - you will learn how to provide them greater value. What do they need from you?

Key Actions for Week 3:

- ☐ Execute your test cases independently
- ☐ Become proficient at but writing
- ☐ Participate actively in at least one team meeting
- ☐ Schedule informal coffee (or tea or water) chats with 2-3 colleagues - learn what they know that you don't

Week 4: Building Confidence

Focus Areas:

- **Independent Work**: Volunteer for small tasks independently if you can create the bandwidth for yourself
- **Quality Focus**: Demonstrate attention to detail in your work - spellcheck, format, include all critical information
- **Proactive Learning**: Identify areas where you need additional knowledge - and seek out a solution
- **Feedback Integration**: Show that you can learn from feedback and improve

Daily Activities:

- Take ownership of specific test cases or small testing areas
- Begin contributing to team discussions with questions and observations

- Seek feedback on your work and implement suggestions
- Start identifying process improvements or areas of confusion - this is information for you for now

Key Actions for Week 4:

- ☐ Complete independent testing assignments
- ☐ Receive positive feedback on at least one piece of work
- ☐ Identify 2-3 areas where you want to develop deeper expertise
- ☐ Schedule your first formal check-in with your manager if this hasn't happened already

Month 1 Success Indicators

By the end of your first month, you should:

- Feel comfortable navigating your work environment and tools
- Understand your team's basic processes and procedures
- Be confident executing test cases and writing bug reports independently
- Established positive working relationships with immediate team members
- Identified your learning priorities for the coming months

Days 31-60: Building Competence and Confidence

Your second month focuses on developing competence in core QA activities and building confidence in your abilities.

Week 5-6: Expanding Your Responsibilities

Focus Areas:

- **Test Case Development**: Be able to create tests on your own
- **Exploratory Testing**: Practice unscripted testing to find unexpected issues - what do the tests NOT cover
- **Cross-Functional Collaboration**: How can best deliver to development, product, project management, etc.
- **Quality Advocacy**: Start thinking about quality from a broader perspective - how does your process impact the final outcome?

Key Activities:

- Take ownership of testing for specific features or components
- Understand in requirement reviews, design discussions, and issue prioritization - participate in these meetings whenever possible and LISTEN
- Determine what gaps your team has that you could work to fill - plan to acquire this expertise
- Review your ideas for process improvements - apply thyem to your own work first and see what creates actual improvement

Weekly Goals:

- ☐ Create test cases for new stories/requirements, seek test type creation that is new to you
- ☐ Find and log bugs through documented tests and exploratory testing - daily
- ☐ Participate in all cross-functional meetings or discussions you have access to
- ☐ Apply at least one process improvement to your own work and track its efficacy

Week 7-8: Demonstrating Value

Focus Areas:

- **Quality Impact**: Document how your testing contributes to product quality - review this with your manager in your regular 1-on-1's
- **Efficiency**: Demonstrate that you can work effectively and meet deadlines - volunteer when you can and then deliver
- **Initiative**: Take on additional responsibilities without being asked - find areas that need extra attention and give it
- **Knowledge Sharing**: Help others learn from your experiences and learn from theirs, compare testing notes, state of the app/product, expectations, etc.

Key Activities:

- Own testing efforts for smaller features or bug fixes
- Contribute to team retrospectives and planning sessions

- Document your learnings to help future new hires and for your own improvement as you progress
- Begin specializing in areas that interest you or match team needs - hwo can you add value? You are now getting paid to learn

Weekly Goals:

- ☐ Successfully own testing for an area of the project or a family of bugs
- ☐ Contribute meaningfully to team planning or retrospective meetings
- ☐ Create documentation that your team can reference for features, processes, etc.
- ☐ Identify a specialization area and begin developing expertise

Month 2 Success Indicators

By the end of your second month, you should:

- Be able to work independently on most routine testing tasks
- Have contributed to improving team processes or documentation
- Feel confident participating in team meetings and discussions
- Have established yourself as a reliable and thorough team member
- Be developing expertise in specific areas of testing or domain knowledge

---

Days 61-90: Establishing Your Reputation

Your third month is about establishing your reputation as a valuable team member and planning for continued growth.

Week 9-10: Taking Ownership

Focus Areas:

- **Leadership**: Take the lead on testing initiatives or projects - whatever is available
- **Contributor**: Become an invaluable tester, test creator, and product expert
- **Strategic Thinking**: Consider long-term quality and how to achieve higher pfoduct quality
- **Stakeholder Relationships**: Build relationships beyond your immediate team - learn how you can provide more value to other teams

Key Activities:

- Own the testing strategy for a significant feature or area
- Represent the QA perspective in planning, review, and prioritization meetings
- Continue to implement improvements to your and your team's work, document processes, product behavior, and test methodology
- Establish achievable career development goals - write them down to keep yourself focused

Weekly Goals:

- [ ] Successfully manage testing for a major feature
- [ ] Provide valuable input in at least 3 strategic discussions
- [ ] Fill in the gaps in your team's docuemntation
- [ ] Create a personal development plan for the next 6 months

Week 11-12: Future Planning

Focus Areas:

- **Performance Review Preparation**: Document your accomplishments and growth - use the numbers you have been tracking
- **Goal Setting**: Establish objectives for your next 90 days and beyond - what is the next set of milestones for you?
- **Skill Development**: Identify advanced skills you want to develop - coordinate with your boss how this can enhance your team
- **Career Planning**: Consider your long-term career trajectory - what do you have to do now to get where you want to be? - write it down

Key Activities:

- Prepare for your 90-day performance review - gather productivity and impact metrics
- Set goals for continued learning and development - know what you want before you talk with your boss about advancement
- Seek feedback from colleagues and stakeholders - ask for insight from them about how you can improve
- Plan your next phase of professional growth - look back to when you started and determine where you want to be next

Weekly Goals:

- [ ] Complete a comprehensive self-assessment of your first 90 days
- [ ] Set specific, measurable goals for the next quarter
- [ ] Receive constructuve feedback from at least 3 different colleagues
- [ ] Create a plan for continued skill development

Month 3 Success Indicators

By the end of your third month, you should:

- Be recognized as a competent, productive, and valuable team member
- Have taken ownership of significant testing responsibilities
- Be contributing to team strategy and process improvements
- Have clear goals for your continued professional development
- Feel confident in your ability to succeed in your QA career

---

The Top 5 Challenges You'll Face

Based on my experience training hundreds of QA professionals, here are the five most common challenges new testers face and how to address them.

Challenge #1: Lack of Respect

The Problem: Some team members may not understand or value the role of Quality Assurance, viewing testers as obstacles rather than contributors.

Why It Happens:

- Misunderstanding of what professional Quality Assurance involves
- Previous bad experiences with ineffective testers
- Pressure to ship products quickly and deprioritzing anything that "slows down" the release process
- Lack of awareness of business value that testing provides and the risk of not prioritizing it

How to Address It:

- **Demonstrate Value**: Show *how* your testing prevents customer issues and saves money
- **Be Solution-Oriented**: When you find problems, suggest solutions when possible, cite requirements, propose alternateves
- **Communicate Impact**: Explain the business consequences of quality issues not just that things aren't working
- **Build Relationships**: Invest time in understanding others' perspectives and challenges - can you deliver in a way that makes their job easier?
- **Stay Professional**: Maintain a positive attitude even when facing resistance - this builds your resistance and refocuses on your future, not your frustration

Practical Tips:

- Focus on facts and data rather than opinions and beliefs
- Celebrate team successes, not your individual achievements

- Learn about the business impact of quality issues so you can better quantify them monetarily
- Volunteer for cross-functional projects to build relationships

Challenge #2: Ignorance About QA

The Problem: Colleagues may have misconceptions about what QA professionals do and how they contribute to product success.

Why It Happens:

- Limited exposure to professional QA practices
- Confusion between QA and other roles (customer support, user research)
- Outdated perceptions of testing as a "low-skill" activity
- Lack of understanding of modern QA methodologies

How to Address It:

- **Educate Through Example**: Show professional QA practices in action and demonstrate their impact
- **Share Knowledge**: Explain your testing approach and reasoning - and get input about how to improve them
- **Highlight Expertise**: Demonstrate the skill and knowledge required for effective testing - **show** how you help
- **Connect to Business Goals**: Show how quality assurance activities (testing and processes) support company objectives

Practical Tips:

- Write clear, detailed bug reports that demonstrate analytical thinking and comprehension of product requirements
- Explain your testing strategy and rationale in team meetings to help others understand and gather input for any omissions
- Give presentations (brown bag lunch, tech talk) about Qualtiy Assurance processes
- Commalborate with other teams about project process, objectivesm, improvements, and deliverables - work together

Challenge #3: Position of QA in the Development Process

The Problem: QA may be positioned as a bottleneck or afterthought rather than an integral part of the development process.

Why It Happens:

- Traditional "waterfall" thinking where testing happens at the end
- Pressure to ship features quickly may leave insufficient time for adequate testing
- Lack of integration between development and QA processes
- Insufficient QA involvement in planning and design

How to Address It:

- **Get Involved Early**: Participate in planning so that test efrfort estimation can be accurately incorporated into the project timeline
- **Shift Left**: Advocate for testing activities throughout the development cycle as early as possible

- **Collaborate Closely**: Work with developers to prevent issues rather than just finding them after they've been created
- **Demonstrate Agility**: Show that QA can work effectively in fast-paced environments - be flexible **and** effective

Practical Tips:

- Attend requirement reviews, design meetings, planning, scoping and timeline meetings
- Provide input on testability and level of effort during feature planning
- Work with developers to create better testing environments
- Advocate for automated testing to speed up feedback cycles

Challenge #4: Lack of Tools and Resources

The Problem: Inadequate tools, environments, or resources that make effective testing difficult or impossible.

Why It Happens:

- Budget constraints or misplaced priorities
- Lack of understanding of QA tool requirements
- Technical debt or legacy systems
- Insufficient investment in testing infrastructure

How to Address It:

- **Document Impact**: Show how tool limitations affect quality and productivity - and how that could be different

- **Propose Solutions**: Research and recommend specific tools or improvements - be cost-concious in your recommendations
- **Build Business Case**: Connect tool investments to business outcomes - show how a tool can increase efficacy, productivity, and coverage
- **Start Small**: Implement improvements incrementally rather than requesting major overhauls - start small, with your own work

Practical Tips:

- Create a prioritized list of tool and resource needs - what will provide the biggest impact?
- Research free or low-cost alternatives to expensive tools - companies prefer not so spend money so be cost-concious
- Collaborate with IT or DevOps teams to improve testing environments - everybody wins
- Track and report on how tool limitations impact your work - and forecast cost and outcome of improvements

Challenge #5: Minuscule Margin for Error

The Problem: The expectation that the QA team will catch every possible issue, and blamed (solely) for bugs found in production.

Why It Happens:

- Unrealistic expectations about what testing can achieve - timelines, resourcing, prioritization
- Lack of understanding of testing limitations and trade-offs
- Pressure from customers or stakeholders for perfect quality

- Insufficient communication about testing coverage and risks

How to Address It:

- **Set Realistic Expectations**: Educate stakeholders about testing limitations - often a math problem of "x" tester hours vs. "y" hours needed for full coverage
- **Communicate Risk**: Clearly explain what has and hasn't been tested and why - this can be mitigated with proper prioritization when planning
- **Focus on High-Impact Issues**: Prioritize testing efforts on the most critical areas - highest liklihood of issue and highest severity of issue
- **Document Coverage**: Maintain clear records of testing scope and results - measure all input (effort, methods, hours) and output (results, decisions)

Practical Tips:

- Create risk assessments for each release and work crossfunctionally to get buy-in
- Communicate testing coverage and limitations clearly so that everyone is prepared going in and as you progress
- Focus on preventing the most impactful issues - these are the best ROI for your testing efforts
- Build relationships with customer support to understand real-world quality issues so you can fine-tune your testing over time

---

Overcoming Common Obstacles

Here are practical strategies for overcoming the most common obstacles new QA professionals face.

Obstacle: Feeling Overwhelmed by Information

Strategy: Create a structured learning plan

- Break down complex topics into smaller, manageable pieces
- Focus on one new concept or tool at a time
- Use your learning journal to track progress and insights
- Ask for help when you need it - everyone expects new hires to have questions

Obstacle: Imposter Syndrome

Strategy: Focus on growth rather than perfection

- Remember that everyone was new once
- Celebrate small wins and progress
- Seek feedback regularly to understand your actual performance and work to improve
- Connect with other QA professionals who can share their experiences

Obstacle: Difficulty Building Relationships

Strategy: Be genuinely interested in others' work

- Ask colleagues about their roles and challenges and how they handle them
- Offer help when you can, even in small ways
- Participate in team social activities and informal conversations
- Show genuine appreciation for others' expertise and contributions

Obstacle: Technical Knowledge Gaps

Strategy: Identify and address specific gaps systematically

- Make a list of technical concepts you need to understand
- Find learning resources (courses, documentation, mentors)
- Practice new skills in low-risk environments
- Don't be afraid to admit when you don't know something

Obstacle: Unclear Expectations

Strategy: Proactively seek clarity

- Ask specific questions about expectations and success criteria
- Request examples of excellent work from previous team members so you have a target
- Schedule regular check-ins with your manager for feedback and expectations
- Document your understanding and confirm it with stakeholders

---

Building Relationships That Matter

Success in QA depends heavily on your ability to build effective working relationships across the organization.

Key Relationships to Develop

1. Your Manager

- Schedule regular one-on-one meetings
- Be proactive in communicating progress and challenges
- Ask for feedback and act on it
- Discuss your career goals and development needs

2. Development Team Members

- Learn about their work and challenges
- Collaborate on preventing issues, not just finding them
- Respect their expertise while advocating for quality
- Build trust through reliable, professional interactions

3. Product Managers

- Understand business requirements and user needs
- Provide input on feature testability and quality risks
- Help translate technical issues into business impact
- Participate in product planning and prioritization discussions

4. Other QA Team Members

- Learn from their experience and expertise
- Share knowledge and collaborate on best practices
- Support each other during challenging projects
- Build a community of practice within your organization

5. Customer Support and Success Teams

- Learn about real-world quality issues and user feedback
- Understand how quality problems impact customers
- Collaborate on reproducing and resolving customer-reported issues
- Use their insights to improve your testing approach

Relationship Building Strategies

Be Genuinely Helpful: Look for opportunities to assist others, even in small ways.

Listen Actively: Show genuine interest in others' perspectives and challenges. You don't have to solve them, listen.

Communicate Clearly: Be concise, accurate, and professional in all interactions.

Follow Through: Always do what you say you'll do, when you say you'll do it.

Show Appreciation: Acknowledge others' contributions and expertise.

Stay Positive: Maintain a constructive attitude, even during difficult situations.

Your 90-Day Success Checklist

Use this checklist to track your progress and ensure you're hitting key milestones during your first 90 days.

Days 1-30: Foundation

- ☐ Completed all administrative setup and training requirements
- ☐ Met with all immediate team members and key stakeholders
- ☐ Gained access to all necessary tools and systems
- ☐ Reviewed existing documentation and test cases
- ☐ Executed first test cases under supervision
- ☐ Wrote first bug reports and received feedback
- ☐ Participated actively in team meetings
- ☐ Established learning goals and priorities
- ☐ Created a system for tracking new knowledge and insights
- ☐ Received constructuve feedback from manager and colleagues

Days 31-60: Building Competence

- ☐ Worked independently on routine testing tasks

- [ ] Created original test cases based on requirements or stories
- [ ] Found and reported legitimate bugs through exploratory testing
- [ ] Participated in cross-functional meetings and discussions
- [ ] Contributed to team process improvements
- [ ] Began developing expertise in specific areas
- [ ] Built positive relationships with developers, project managers, designers, and product managers
- [ ] Demonstrated reliability and attention to detail
- [ ] Sought and acted on feedback from multiple sources
- [ ] Identified areas for continued learning and development

Days 61-90: Establishing Reputation

- [ ] Successfully owned testing efforts for significant features or areas
- [ ] Contributed meaningfully to team strategy and planning
- [ ] Represented QA perspective in stakeholder meetings
- [ ] Created documentation or process(es) that help(s) the team
- [ ] Received recognition for quality contributions
- [ ] Set clear goals for continued professional development
- [ ] Built relationships beyond immediate team
- [ ] Demonstrated initiative and proactive problem-solving
- [ ] Prepared comprehensive self-assessment for performance review

Overall Success Indicators

- [ ] Feel confident in your ability to perform core QA tasks
- [ ] Have established yourself as a reliable and valuable team member
- [ ] Understand your company's products and quality standards
- [ ] Built positive working relationships across the organization
- [ ] Identified your strengths and areas for continued growth
- [ ] Have a clear plan for your next phase of development
- [ ] Received positive feedback in your 90-day review
- [ ] Feel excited about your future in QA

Conclusion

Your first 90 days as a software tester are just the beginning of what can be a rewarding and impactful career. By following the structured approach outlined in this guide, you can establish yourself as a valuable team member while building the foundation for long-term success.

Remember that everyone's journey is different, and it's normal to face challenges and setbacks along the way. The key is to maintain a growth mindset, seek feedback actively, and focus on continuous improvement.

The relationships you build and the reputation you establish during these first 90 days can serve you throughout your career. Invest in them wisely, and always remember that your role as a QA professional is to advocate for quality and help create products that truly serve users' needs.

Welcome to the QA profession - your journey toward becoming an elite software tester has just begun!

About Eochair

This guide is free. The team behind it is building the tool QA professionals wish they'd had from day one.

Most testing tools make *you* work for *them* — you bend your process around the tool, maintain traceability by hand, and watch requirements, tests, and issues drift into separate silos until nobody remembers what the feature was even supposed to do.

> *" "Automating Jira is the absolute worst programming experience I've ever had."*
>
> *"Starting to hit the limits of how we're handling traceability without everything breaking. Losing my mind basically.""*

A tool should work for you, not the other way around. **Eochair** keeps your requirements, tests, and issues in one place and links them automatically — so your spec stays alive instead of evaporating the moment a story gets closed.

Built by the Eochair team, led by a 30-year QA veteran — the same person who wrote this guide.

Eochair is launching soon. Join the waitlist → guides.eochair.com

---

Get the full free series & early access — scan, or visit **guides.eochair.com**